Regresión Lineal simple

September 17, 2020

1 Regresión lineal simple

1.1 El paquete statsmodel para regresión lineal

[17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
%matplotlib inline
```

[18]:
```
data = pd.read_csv("D:\Rodolfo\Clases\Mineria/Advertising.csv")
```

[19]:
```
data.head()
```

[19]:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

[20]:
```
lm = smf.ols(formula="Sales~TV",data = data).fit()
```

[21]:
```
lm.params
```

[21]:
```
Intercept    7.032594
TV           0.047537
dtype: float64
```

Modelo lineal predictivo : * Sales = 7.032594 + 0.047537 * TV

[22]:
```
lm.pvalues
```

[22]:
```
Intercept    1.406300e-35
TV           1.467390e-42
dtype: float64
```

[23]:
```
lm.rsquared
```

[23]: 0.611875050850071

[24]:
```
lm.rsquared_adj
```

[24]: 0.6099148238341623

[26]:
```
lm.summary()
```

[26]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.612
Model:                            OLS   Adj. R-squared:                  0.610
Method:                 Least Squares   F-statistic:                     312.1
Date:                Thu, 17 Sep 2020   Prob (F-statistic):           1.47e-42
Time:                        09:48:48   Log-Likelihood:                -519.05
No. Observations:                 200   AIC:                             1042.
Df Residuals:                     198   BIC:                             1049.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      7.0326      0.458     15.360      0.000       6.130       7.935
TV             0.0475      0.003     17.668      0.000       0.042       0.053
==============================================================================
Omnibus:                        0.531   Durbin-Watson:                   1.935
Prob(Omnibus):                  0.767   Jarque-Bera (JB):                0.669
Skew:                          -0.089   Prob(JB):                        0.716
Kurtosis:                       2.779   Cond. No.                         338.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

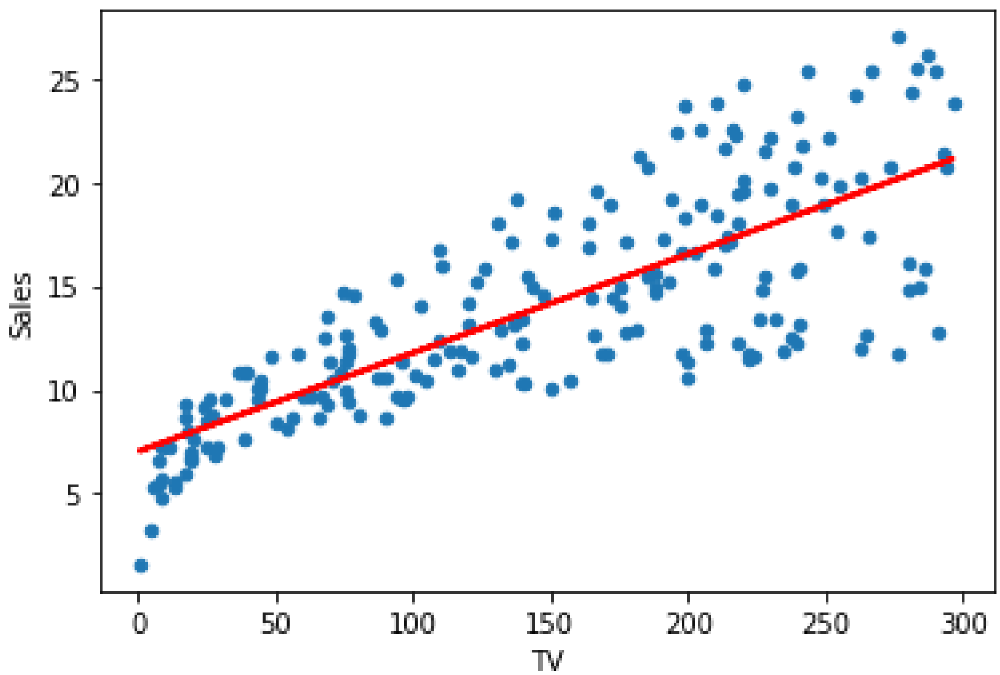
[27]:
```
sales_pred = lm.predict(pd.DataFrame(data["TV"]))
sales_pred
```

[27]:
```
0      17.970775
1       9.147974
2       7.850224
3      14.234395
```

```
4      15.627218
         ...
195     8.848493
196    11.510545
197    15.446579
198    20.513985
199    18.065848
Length: 200, dtype: float64
```

[28]:
```
data.plot(kind = "scatter" ,x ="TV", y ="Sales")
plt.plot(pd.DataFrame(data["TV"]),sales_pred,c="red", linewidth=2)
```

[28]:
```
[<matplotlib.lines.Line2D at 0x181d1ee1088>]
```



[29]:
```
data["sales_pred"] = 7.032594 + 0.047537 * data["TV"]
```

[31]:
```
data["RSE"] = (data["Sales"]-data["sales_pred"])**2
```

[35]:
```
SSD = sum(data["RSE"])
SSD
```

[35]:
```
2102.5305838896525
```

[36]:
```
RSE = np.sqrt(SSD/(len(data)-2))
RSE
```

[36]: 3.258656369238098

[37]:
```
sales_m = np.mean(data["Sales"])
sales_m
```

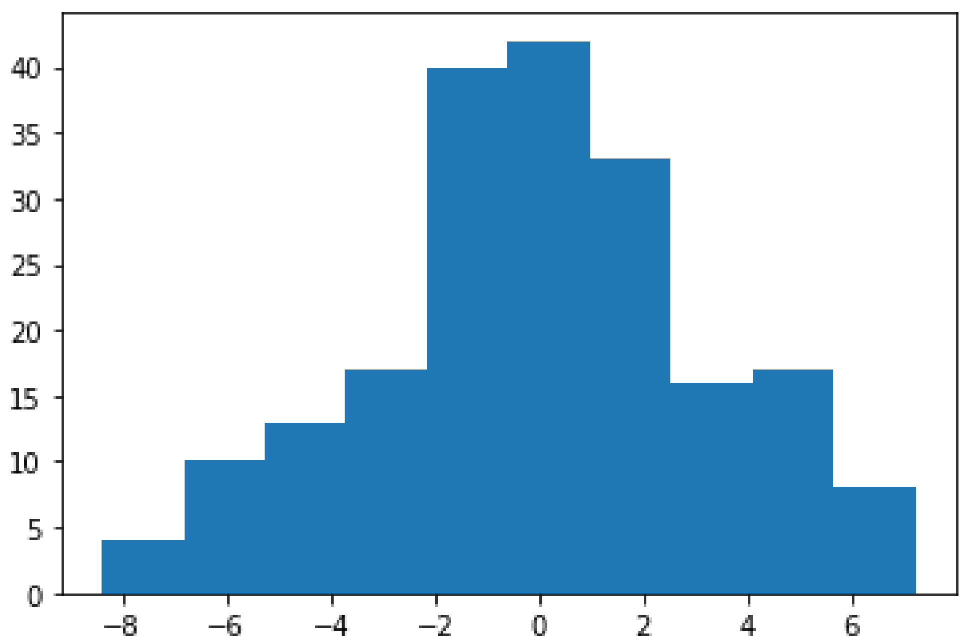
[37]: 14.022500000000003

[38]:
```
RSE / sales_m
```

[38]: 0.2323876890168014

[39]:
```
plt.hist((data["Sales"]-data["sales_pred"]))
```

[39]:
```
(array([ 4., 10., 13., 17., 40., 42., 33., 16., 17.,  8.]),
 array([-8.3860819 , -6.82624404, -5.26640618, -3.70656832, -2.14673046,
        -0.5868926 ,  0.97294526,  2.53278312,  4.09262098,  5.65245884,
         7.2122967 ]),
 <a list of 10 Patch objects>)
```



[ ]:
```
```